

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 05-143410

(43)Date of publication of application : 11.06.1993

(51)Int.Cl.

G06F 12/00

G06F 15/40

(21)Application number : 03-305930

(71)Applicant : HITACHI LTD

(22)Date of filing : 21.11.1991

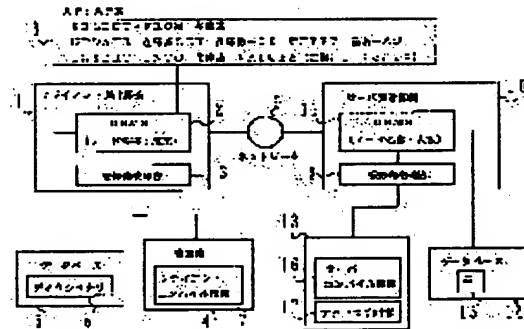
(72)Inventor : TANABE KIYOSHI

(54) ACCESS CONTROL METHOD FOR DISTRIBUTED DATA BASE

(57)Abstract:

PURPOSE: To accelerate distributed data base access by compiling respective applied programs (AP) in advance, generating an order to access a data base and registering the order to reduce time required for compile in the case of execution.

CONSTITUTION: A client side computer 1 is composed of a data base 5, DBMS 2 to manage the data base 5, registration list managing part 3 to manage a registration list 4, and AP 8 describing the various relational collation language sentences of relational data base languages. A server side computer 10 is composed of a data base 14, DBMS 11 to manage the data base 14, and registration list managing part 12 to manage a registration list 13. The AP 8 to access the respective data bases 5 and 14 is compiled in advance, the registration list managing parts 3 and 12 register the order for the AP 8 to access the data bases on the registration lists 4 and 13 together with compile information, and when executing the AP 8, the data bases are accessed at high speed by reading the compile information from the registration lists 4 and 13.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-143410

(43)公開日 平成5年(1993)6月11日

(51)Int.Cl.⁵G 0 6 F 12/00
15/40

識別記号

5 1 3 J 7832-5B
5 0 0 M 7060-5L

庁内整理番号

F I

技術表示箇所

審査請求 未請求 請求項の数1(全10頁)

(21)出願番号

特願平3-305930

(22)出願日

平成3年(1991)11月21日

(71)出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72)発明者 田辺 清

神奈川県横浜市戸塚区戸塚町5030番地 株

式会社日立製作所ソフトウェア開発本部内

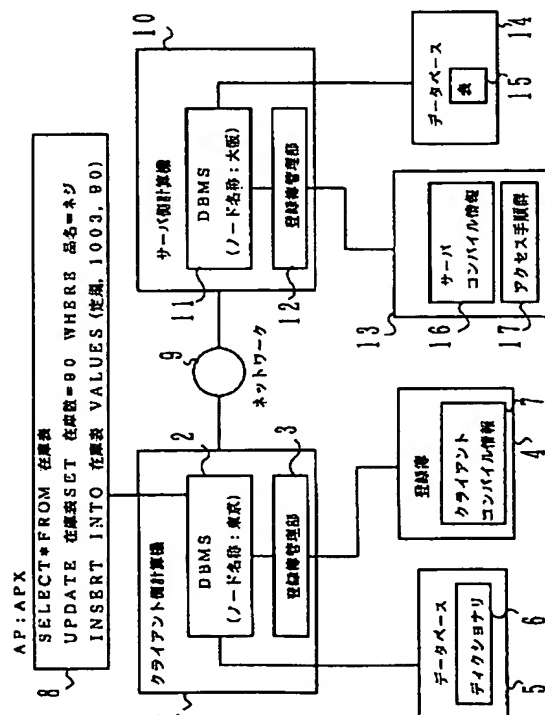
(74)代理人 弁理士 小川 勝男

(54)【発明の名称】 分散データベースのアクセス制御方法

(57)【要約】

【目的】 実行時におけるコンパイルに要する時間を削減し、分散データベースアクセスを高速化する。

【構成】 分散データベースにおいて、それぞれのデータベースをアクセスする応用プログラム(AP)を、予めコンパイルし、このAPのデータベースをアクセスする手順を、コンパイル情報と共に、登録簿管理部により、登録簿に登録しておき、APの実行時には、登録簿管理部で登録簿よりコンパイル情報を読みだして、高速にアクセスする。



【特許請求の範囲】

【請求項1】 複数の応用プログラムからアクセスされるデータベースと、上記応用プログラムからの上記データベースへの処理要求に基づき、該データベースを操作するデータベース管理システムと、該データベース管理システムで、上記応用プログラムをコンパイルして得た情報を登録する登録手段と、該登録手段への上記情報の登録処理および該登録手段に登録した情報の読み出し処理を行なう登録簿管理手段とを有する複数のノードを、ネットワークで結合してなり、任意の上記応用プログラムで、任意のノードから任意のノードのデータベースへのアクセスを行なう分散データベースのアクセス制御方法において、上記ノードのそれぞれは、予め、該それぞれのノードの上記データベース管理システムが上記複数の応用プログラムをコンパイルして得た上記データベースへのそれぞれのアクセス手順を、それぞれの応用プログラム対応に、上記登録手段に格納し、上記応用プログラム実行時に、上記ノードのそれぞれのデータベース管理システムは、上記登録簿管理手段が上記登録手段から読み出した上記応用プログラム対応のデータベースへのアクセス手順に基づき、上記データベースのアクセス操作を行なうことを特徴とする分散データベースのアクセス制御方法。

【発明の詳細な説明】**【0001】**

【産業上の利用分野】 本発明は、応用プログラム（以下、APと記載：Application Program、アプリケーションプログラム）からの分散データベースへのアクセス制御方法に係わり、特に、アクセス処理を高速化するのに好適な分散データベースのアクセス制御方法に関するものである。

【0002】

【従来の技術】 従来、分散データベースにおいては、各利用者が、接続されたそれぞれのデータベースやAPの違いを意識せずに利用できる必要があり、一般には、いわゆるネットワークディクショナリと呼ばれる情報の登録簿を用いて、資源の管理情報を1元的に管理することにより可能としている。ネットワークディクショナリには、各ノードの接続関係を示すネットワーク情報や、各ノードに分散配置されているデータベースの情報である分散データベース情報、また、各ノードで処理分担するプログラム情報である応用プログラム情報や、分散処理システムを使用できる利用者の管理情報である利用者情報、そして、データベース管理システム（以下、DBMSと記載、DataBase Management System）がコンパイルして得た情報などが格納される。

【0003】 そして、資源の管理やアクセスを統一的行なうために、いわゆる統合化アプローチなどによる仮想化技術が用いられている。統合化アプローチとは、他の計算機の各種資源を自システムの資源にマッピングし

て、あたかも、自システムの資源として扱うように仮想化する方法である。このような異種分散データベースでは、分散されたデータベース間で、データの構造や操作言語が異なり、これらの異なるシステムの統合には、登録簿（ネットワークディクショナリ）に基づくデータモデル変換やデータ変換、また、プログラム言語変換やデータベース操作言語変換などの変換技法を使用して対処する必要がある。そのために、データモデル変換やデータ変換を行なうことにより、分散データベースのアクセスに余分な時間を要していた。

【0004】 このようなデータ変換に伴うアクセス時間の低速化に対処するための従来技術として、例えば、特開平02-69871号公報に記載のように、データベース問合せ（アクセス要求）が入力されると、DBMSは、データベース問合せデータを編集する際に、データ変換の要不要を判断して、不要なデータ変換を排除することにより、問合せ処理を高速化するものがある。

【0005】 しかし、上述の技術では、データベースへのアクセス手順のコンパイルに係わる処理時間のロスに対する考慮がなされていない。すなわち、DBMSは、ネットワークを用いて各計算機に分散されたデータベースをアクセスするAPの実行時、分散された各計算機でデータベースをアクセスする手順をコンパイルして生成した後に、このデータベースアクセス手順に従い、データベースをアクセスする。そのために、APの実行時間は、コンパイルに要する時間分、余分に長くなっていた。

【0006】

【発明が解決しようとする課題】 解決しようとする問題点は、従来、分散データベースにおいて、DBMSは、AP実行時に、ネットワークを用いて各計算機に分散されたデータベースをアクセスする手順を生成するための時間がかかり、高速にデータベースをアクセスすることができない点である。本発明の目的は、これら従来技術の課題を解決し、予め、それぞれのAPをコンパイルして、データベースをアクセスする手順を生成して登録しておき、AP実行時のアクセス手順の生成を不要とし、アクセス処理の高速化を可能とする分散データベースのアクセス制御方法を提供することである。

【0007】

【課題を解決するための手段】 上記目的を達成するため、本発明の分散データベースのアクセス制御方法は、
（1）複数の応用プログラムからアクセスされるデータベースと、応用プログラムからのデータベースへの処理要求に基づき、このデータベースを操作するデータベース管理システムと、このデータベース管理システムで、応用プログラムをコンパイルして得た情報を登録する登録簿と、この登録簿への情報の登録処理およびこの登録簿に登録した情報の読み出し処理を行なう登録簿管理部とを有する複数のノードを、ネットワークで結合してな

り、任意の応用プログラムで、任意のノードから任意のノードのデータベースへのアクセスを行なう分散データベースのアクセス制御方法において、ノードのそれぞれは、予め、それぞれのノードのデータベース管理システムが複数の応用プログラムをコンパイルして得たデータベースへのそれぞれのアクセス手順を、それぞれの応用プログラム対応に、登録簿に格納し、応用プログラム実行時に、ノードのそれぞれのデータベース管理システムは、登録簿管理部が登録簿から読み出した応用プログラム対応のデータベースへのアクセス手順に基づき、データベースのアクセス操作を行なうことを特徴とする。

【0008】

【作用】本発明においては、APを管理するDBMSは、APコンパイル時に、APのデータベースアクセス言語を仮想記憶に受け取り、このデータベース言語が、分散されたデータベースをアクセスするものであれば、該当するデータベースを管理するDBMSに、このデータベース言語を送るため、ネットワークに対しメッセージとして送信要求を行う。ネットワークは送信要求に従い、該当するDBMSにデータベース言語を送る。該当するDBMSは、ネットワークにより送られたデータベース言語を仮想記憶に受け取り、コンパイルを行う。そして、登録簿管理部は、コンパイルにより得たデータベースをアクセスする手順を、登録簿に登録する。そして、APを管理するDBMSは、APの実行時、このAP中のデータベース言語を仮想記憶に受け取り、このデータベース言語がコンパイルされていて、かつ、分散されたデータベースをアクセスする場合、該当するDBMSへ送るため、ネットワークに対しメッセージとして送信要求を行う。ネットワークは、送信要求に従い、該当するDBMSにデータベース言語を送る。該当するDBMSが、ネットワークにより送られてきたデータベース言語を仮想記憶に受け取ると、登録簿管理部で、登録簿を参照し、データベース言語がコンパイルされていることを確認し、データベースをアクセスする手順を仮想記憶に読み出す。そして、DBMSは、登録簿管理部が読み出したデータベースのアクセス手順に従いデータベースをアクセスする。このことにより、AP実行時のコンパイル処理時間の削除ができ、高速なデータベースのアクセスが可能となる。

【0009】

【実施例】以下、本発明の実施例を、図面により詳細に説明する。図1は、本発明を施したリレーショナル型の分散データベースの本発明に係わる構成の一実施例を示すブロック図である。本実施例の分散データベースは、クライアント側計算機1とサーバ側計算機10とを、ネットワーク9で接続することにより構成されている。そして、クライアント側計算機1は、データベース5、および、このデータベース5を管理するDBMS2と、登録簿4を管理する登録簿管理部3と、リレーショナルデ

ータベース言語の各種SQL (Structured Query Language、関係型照会言語) 文を記述したAP8とからなる。同様に、サーバ側計算機10は、データベース14、および、このデータベース14を管理するDBMS11と、登録簿13を管理する登録簿管理部12とからなる。

【0010】このような構成の分散データにおいて、クライアント側計算機1のDBMS2は、クライアント側でAP8のコンパイル時、データベース5のディクショナリ6を参照することにより、AP8中のSQL文でアクセスするデータベース中の表が、サーバ側計算機10にあるか否かを判断する。また、サーバ側計算機10では、AP8の実行時のアクセス手順により、データベース14中の表15をアクセスする。そして、クライアント側計算機1の登録簿管理部3は、クライアント側でのAP8のコンパイル時、DBMS2の指示に従い、AP8のクライアントコンパイル情報7を、登録簿4に登録し、そして、AP8の実行時に、DBMS2の指示に従い、登録簿4から、図示していない仮想記憶に、クライアントコンパイル情報7を読み込む。また、サーバ側計算機10の登録簿管理部12は、サーバ側でのAP8のコンパイル時に、DBMS11の指示に従い、AP8のサーバコンパイル情報16、および、データベースのアクセス手順群17を登録簿13に登録し、そして、AP8の実行時に、DBMS11の指示に従い、登録簿13より、サーバコンパイル情報16、および、データベースのアクセス手順群17中の該当するアクセス手順を仮想記憶に読み込む。このように、予め、登録簿13に、AP8のデータベースのアクセス手順を、アクセス手順群17として登録しておくことにより、サーバ側計算機10では、AP8の実行時におけるデータベース14中の表15へのアクセスを、高速に行なうことができる。以下、本実施例の分散データベースに係わる構成および動作を、図2～図9を用いてさらに詳しく説明する。

【0011】図2は、図1におけるクライアント側計算機のデータベース内のディクショナリの構成の一実施例を示す説明図である。本実施例のディクショナリ6は、図1のAP8がアクセスする表の「表名称」と、表が存在するノードの「ノード名称」との対応関係を、図1のAP8のコンパイルまでに事前登録したものである。例えば、本実施例では、「在庫表」が「大阪」のノードに存在する対応関係が、データとして登録されている。

【0012】図3は、図1におけるクライアント側計算機の登録簿内のクライアントコンパイル情報の構成の一実施例を示す説明図である。本実施例のクライアントコンパイル情報7は、SQL文によりアクセスする表が存在する「実行先ノード名称」と「AP名称」、および、「アクセス手順番号」(AP中のSQL文に対応する番号)と「アクセス手順相対番地」を、図1のAP8のコンパイル時に、図1におけるクライアント側の登録簿管

理部3により登録されたものである。

【0013】図4は、図1におけるサーバ側計算機のデータベース中の表の構成の一実施例を示す説明図である。本実施例の表は、在庫表18であり、この在庫表18は、ネジやクギなどの「品名」、それぞれの品名に対応する「品番」および「在庫数」からなる表で、データが数件あるものとする。

【0014】図5は、図1におけるサーバ側計算機の登録簿中のサーバコンパイル情報およびアクセス手順群の構成の一実施例を示す説明図である。本実施例のサーバコンパイル情報16は、図1のAP8が存在する「要求元ノード名称」と「AP名称」、および、SQL文をコンパイルし生成されるアクセス手順の識別子としての「アクセス手順番号」と、このアクセス手順の登録場所を示す「アクセス手順相対番地」を、コンパイル処理時において、図1のサーバ側の登録簿管理部12により、図1の登録簿13に登録されたものである。また、本実施例のアクセス手順群17は、図1のAP8をコンパイルすることにより、このAP8中のSQL文毎に、データベースをアクセスするアクセス手順19～21が、図1のDBMS11により生成され、図1の登録簿管理部12により、図1の登録簿13に登録されたものである。

【0015】次に、図6～図9を用い、図1の分散データベース下のAPのコンパイル、および、実行の動作について述べる。図6は、図1におけるクライアント側計算機の本発明に係わる処理動作の一実施例を示すフローチャートである。本実施例は、図1におけるクライアント側計算機1のDBMS2によるAP8のコンパイル動作を示すものであり、DBMSは、まず、図1のAP8を、図1のクライアント側計算機1の仮想記憶に読み込む（ステップ601）。そして、読み込んだAP8を解析して、一つずつSQL文に分解する（ステップ602）。例えば、一つ目のSQL文「SELECT * FROM 在庫表」を、構文上、および、意味上解析し、アクセスする表が在庫表であることを認識する。そして、図2のディクショナリ6から、表名称が在庫表であるデータを仮想記憶に読み込み、在庫表が存在するノードが、大阪（リモート）であることを決定する（ステップ603）。次に、SQL文が、他ノードか否かを判断して（ステップ604）、自ノードのSQL文であれば、自ノードの処理を行う（ステップ610）。しかし、本実施例では、当該SQL文が大阪のノードであることを決定しているため、ネットワークに対し、大阪のノードへ、当該SQL文を一番目の要素に、また、二番目の要素に「自ノードの名称：東京」を、三番目の要素に当該SQL文の「アクセス手順番号：1」を、四番目の要素に当該APのAP名称を、そして、五番目の要素にコンパイル要求を示す情報を、メッセージとして送るように送信要求を行う（ステップ605）。そして、後

述の図7に示すこの大阪のノード（本実施例では、図1のサーバ側計算機10）でのSQLコンパイル処理が終わった後、ネットワークより、メッセージを仮想記憶に受け取る。このメッセージには、一番目の要素にSQL文の「アクセス手順番号：1」が、二番目の要素に「リターンコード」がある。そして、DBMSは、受け取った大阪のノードのSQLコンパイル処理の結果であるリターンコードを判断（ステップ606）する。例えば、リターンコードが「0」であれば、大阪のノードでのSQLコンパイル処理が正常に行なわれたものとして、図1の登録簿管理部3を呼ぶ。この図1の登録簿管理部3は、図1の登録簿4のクライアントコンパイル情報7に、図3に示すように、「実行先ノード名称：大阪」、「AP名称：APX」、および、「アクセス手順番号：1」を、仮想記憶より読みだしてデータとして登録する（ステップ607）。また、リターンコードが「0」以外であれば、DBMSは、エラー処理を行う（ステップ609）。次に、DBMSは、仮想記憶中の図1のAP8を解析して、SQL文が終了しているか判断する（ステップ608）。この判断結果で、SQL文が未だあれば、ステップ602～608を同様に行い、図1のAP8の終了まで繰り返す。このようにして、図1のDBMS2は、図1のAP8のコンパイルを行なう。

【0016】図7は、図1におけるサーバ側計算機の本発明に係わる処理動作の一実施例を示すフローチャートである。本実施例は、図1におけるサーバ側計算機10のDBMS11によるAP8のコンパイル動作を示すものであり、図6のステップ605において、図1に示したクライアント側のDBMS2が、大阪のノードへSQLコンパイル処理要求を行った際の動作である。大阪のノード（本実施例では、図1のサーバ側計算機10）のDBMSは、まず、図1のネットワーク9から、図1のサーバ計算機10の仮想記憶に、メッセージとして、一番目の要素にコンパイルするSQL文「SELECT * FROM 在庫表」を、また、二番目の要素にコンパイル処理要求した「ノード名称：東京」を、三番目の要素に当該SQL文の「アクセス手順番号：1」を、四番目の要素に当該APのAP名称を、そして、五番目の要素にコンパイル要求を示す情報を受け取る（ステップ701）。この五番目の要素で受け取った情報がコンパイル要求であることから、受け取ったSQL文を、構文上、および、意味上解析し、データベースをアクセスする手順を生成する（ステップ702）。そして、アクセス手順が生成出来たか否かを判断する（ステップ703）。ここで、アクセス手順が生成できた場合、図1の登録簿管理部12を呼ぶ。この登録簿管理部は、図5に示すように、サーバコンパイル情報として、「要求元ノード名称：東京」と「AP名称：APX」、「アクセス手順番号：1」、および、「アクセス手順相対番地」を仮想記憶より読みだして、図1の登録簿13のサーバコ

ンパイル情報16に書き込む(ステップ704)。この「アクセス手順相対番地」は、図1の登録簿管理部12が、図5に示すアクセス手順群17に、アクセス手順を書き込む時に記憶する相対番地である。次に、DBMSは、ネットワークに対し、一番目の要素に「SQL文のアクセス手順番号:1」を、二番目の要素に「リターンコード:0」をメッセージとして、クライアントである東京に送信要求して終了する。ステップ703において、アクセス手順が生成出来なかった場合、エラーとして、ネットワーク9に対し、「リターンコード:0」以外の数値をメッセージとして、クライアントである東京に送信要求して終了する(ステップ705)。以下、同様に、クライアント側のSQLコンパイル要求が終了するまでステップ701~705を繰り返し、このようにして、図1のDBMS11は、図1のAP8のコンパイルを行なう。

【0017】次に、図1の分散データベース下のAP8実行を、図8と図9のフローチャートを用いて説明する。図8は、図1におけるクライアント側計算機1のDBMS2によるAP8の実行動作を示すのものであり、DBMSは、利用者が入力した「AP8:APX」の実行要求に従い、まず、図1のクライアント側計算機1の仮想記憶にAP8を読み込み(ステップ801)、図1の登録簿管理部3を呼ぶ。この登録簿管理部は、図1の登録簿4のクライアントコンパイル情報7から、図3に示すAP名称がAPXのデータである「実行先ノード名称:大阪」と、「AP名称:APX」と、「アクセス手順番号:1」を仮想記憶に読み込む(ステップ802)。そして、アクセス手順番号が「1」のSQL文が、リモートの処理か否かを、「実行先ノード名称」を参照して判断する(ステップ803)。図3に示すように、ここでは、「実行先ノード名称」が「大阪」であることから、大阪のノードに対し、実行要求を行う(ステップ804)。この実行要求とは、DBMSが、図1のネットワーク9に対し、一番目の要素に「AP名称:APX」、二番目の要素に「アクセス手順番号:1」、三番目の要素に「要求元ノード名称:東京」、四番目の要素に実行要求である情報をメッセージとして大阪に送るように、送信要求することである。もし、ステップ803において、「実行先ノード名称」が「自ノード:東京」であれば、DBMSは、自ノードのデータベース(図1のデータベース5)をアクセスする(ステップ805)。そして、大阪のノードでのSQL実行が終了した後、図1のAP8が終了するまで、ステップ802~804、805を繰り返す(ステップ806)。このようにして、図1のDBMS2は、図1のAP8を実行する。

【0018】図9は、図1におけるサーバ側計算機の本

発明に係わる処理動作の一実施例を示すフローチャートである。本実施例は、図1におけるサーバ側計算機10のDBMS11によるAP8の実行動作を示すのものであり、DBMSは、まず、図8のステップ804において送信された実行要求を、ネットワーク9を介して、図1のサーバ計算機10の仮想記憶に読み込む(ステップ901)。この実行要求は、一番目の要素に「要求元ノード名称:東京」、二番目の要素に「AP名称:APX」、三番目の要素に「アクセス手順番号:1」、四番目の要素に実行要求である情報がメッセージとして送信されてきたものである。次に、DBMSは、この四番目の要素に受け取った情報が実行要求であることから、図1の登録簿管理部12を呼ぶ。この登録簿管理部は、図1の登録簿13のサーバコンパイル情報16から、図5に示すように、「AP名称:APX」、「アクセス手順番号:1」、「要求元ノード名称:東京」であるデータの「アクセス手順相対番地」を仮想記憶に読み込み、この「アクセス手順相対番地」が示すアクセス手順群17の相対番地から、アクセス手順を仮想記憶に読み込む(ステップ902)。そして、DBMSは、この登録簿管理部が読み込んだアクセス手順に従い、図1のデータベース14中の表15の内の図4に示す在庫表18のデータをアクセスする(ステップ903)。以下、同様にして、クライアント側の実行要求が終了するまでステップ901~903を繰り返し、このようにして、図1のDBMS11は、図1のAP8を実行する。

【0019】以上、図1~図9を用いて説明したように、本実施例の分散データベースでは、分散データベースをアクセスするAPを、予め、コンパイルして、登録簿に、コンパイル情報のみならず、データベースをアクセスする手順を登録する。このことにより、APの実行時には、分散データベース下で、コンパイル処理を行なう必要がなくなり、APの実行時間を短縮することができる。尚、本発明は、図1~図9を用いて説明した実施例に限定されるものではなく、例えば、図1のネットワーク9で、クライアント側計算機1とサーバ側計算機10の他にも、データベースを有する計算機を結合する構成でも良い。

【0020】

【発明の効果】本発明によれば、AP実行時のコンパイルが不要となり、アクセス処理の高速化が可能となり、AP実行の性能を向上させることができる。

【0021】

【図面の簡単な説明】

【図1】本発明を施したリレーショナル型の分散データベースの本発明に係わる構成の一実施例を示すブロック図である。

【図2】図1におけるクライアント側計算機のデータベース内のディクショナリの構成の一実施例を示す説明図である。

【図3】図1におけるクライアント側計算機の登録簿内のクライアントコンパイル情報の構成の一実施例を示す説明図である。

【図4】図1におけるサーバ側計算機のデータベース中の表の構成の一実施例を示す説明図である。

【図5】図1におけるサーバ側計算機の登録簿中のサーバコンパイル情報およびアクセス手順群の構成の一実施例を示す説明図である。

【図6】図1におけるクライアント側計算機の本発明に係わる処理動作の一実施例を示すフローチャートである。

【図7】図1におけるサーバ側計算機の本発明に係わる処理動作の一実施例を示すフローチャートである。

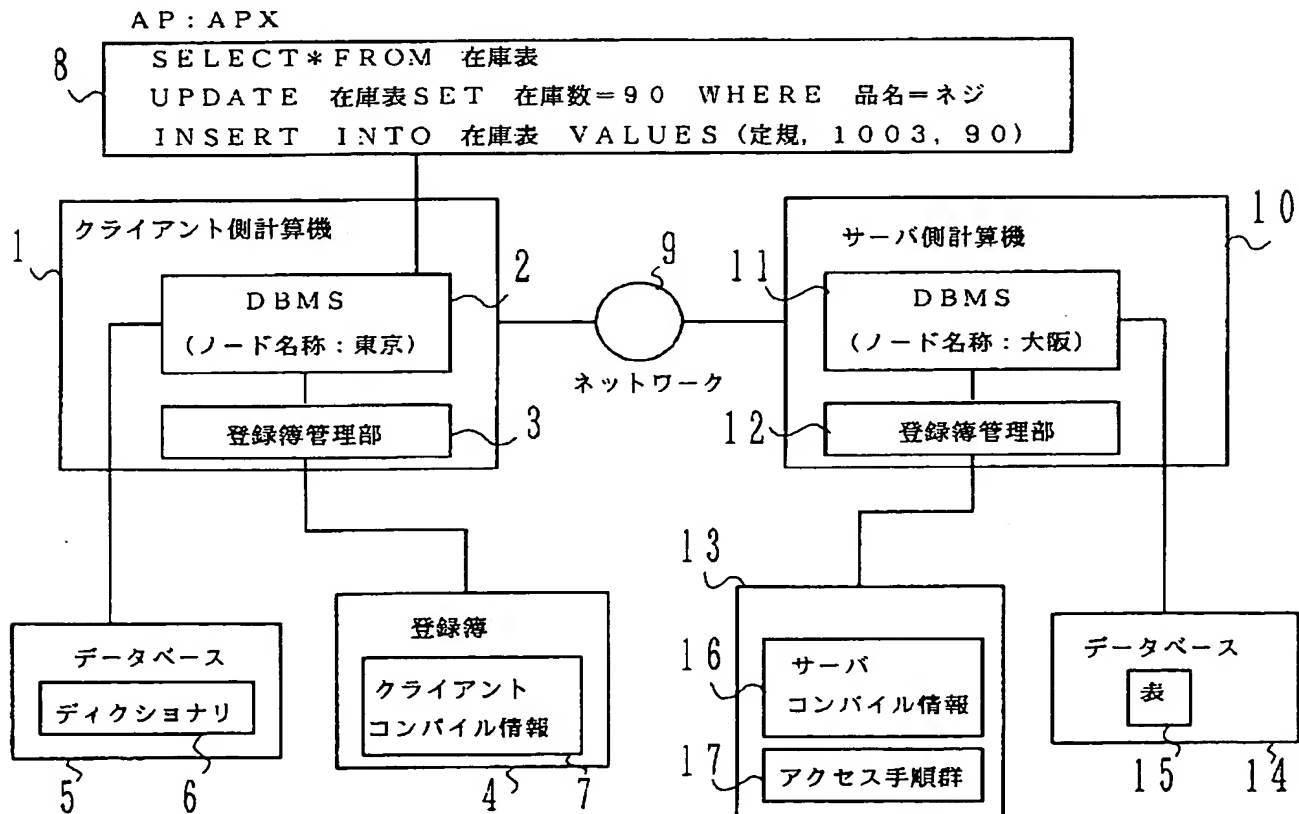
【図8】図1におけるクライアント側計算機の本発明に係わる処理動作の一実施例を示すフローチャートである。

【図9】図1におけるサーバ側計算機の本発明に係わる処理動作の一実施例を示すフローチャートである。

【符号の説明】

- 1 クライアント側計算機
- 2 DBMS
- 3 登録簿管理部
- 4 登録簿
- 5 データベース
- 6 ディクショナリ
- 7 クライアントコンパイル情報
- 8 AP
- 9 ネットワーク
- 10 サーバ計算機
- 11 DBMS
- 12 登録簿管理部
- 13 登録簿
- 14 データベース
- 15 表
- 16 サーバコンパイル情報
- 17 アクセス手順群
- 18 在庫表
- 19～21 アクセス手順

【図1】



【図2】

ディクショナリ	
表名称	ノード名称
在庫表	大阪
書籍表	名古屋
...	...

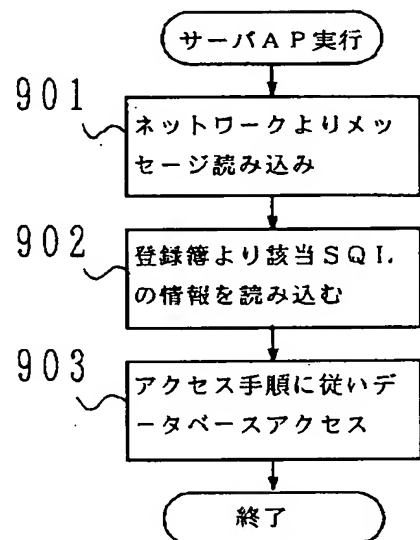
【図4】

在庫表		
品名	品番	在庫数
ネジ	1001	100
クギ	1002	50
...

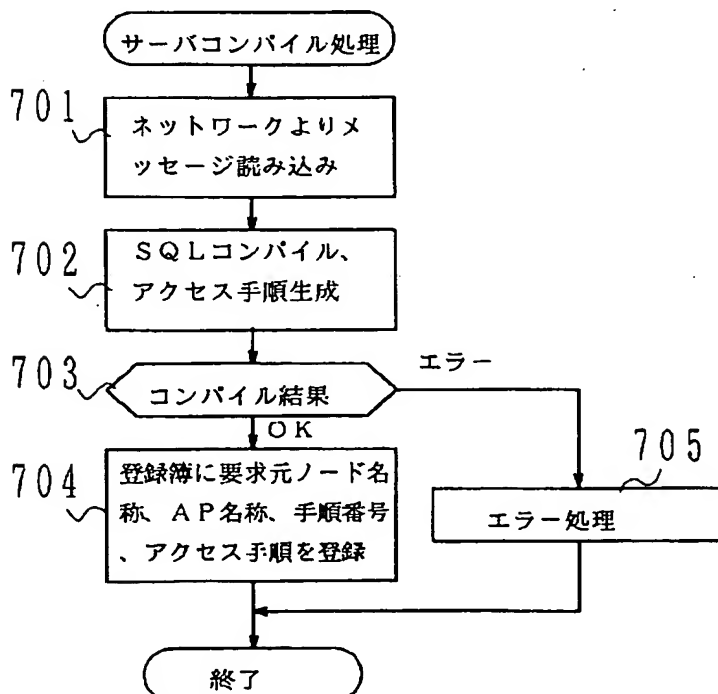
【図3】

クライアントコンパイル情報			
実行先ノード 名称	AP名称	アクセス手順 番号	アクセス手順 相対番地
大阪	APX	1	-
大阪	APX	2	-
大阪	APX	3	-
...

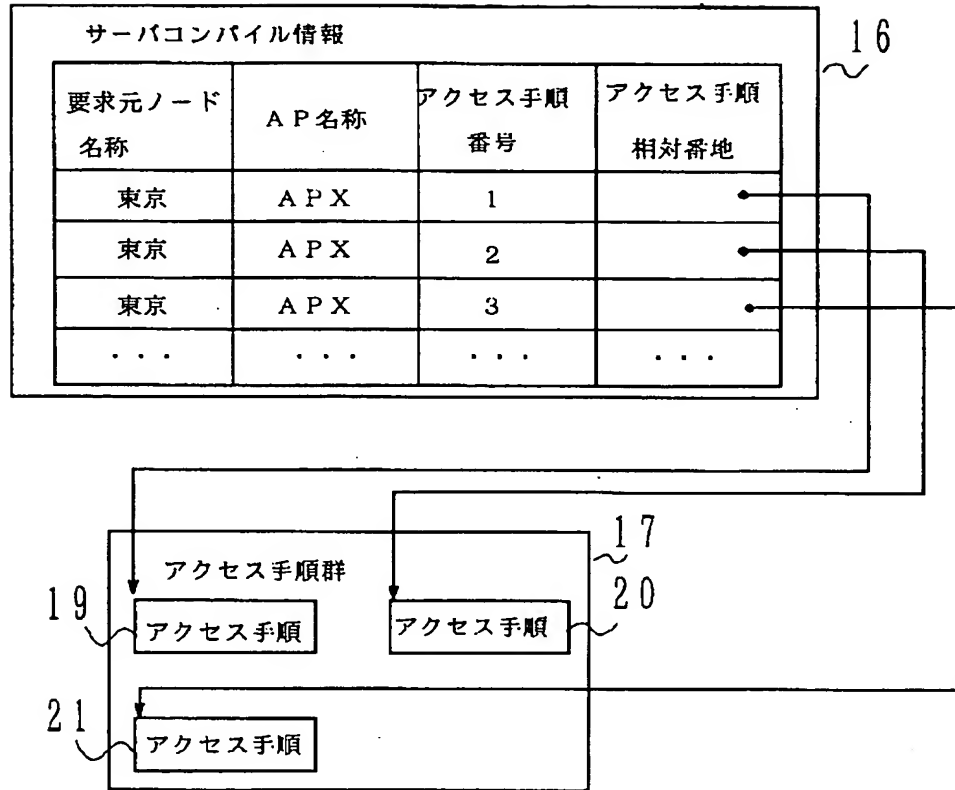
【図9】



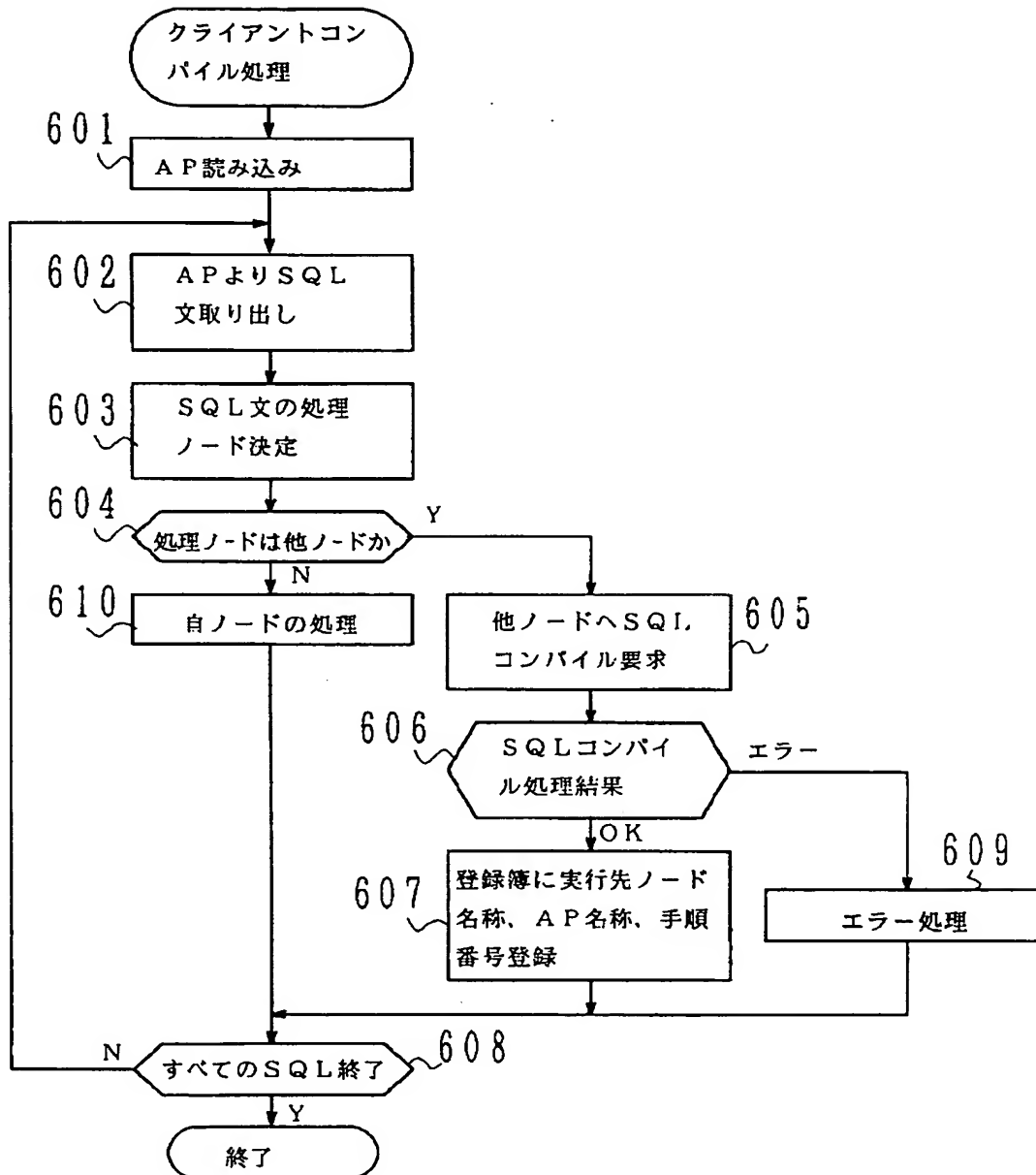
【図7】



【図5】



【図6】



【図8】

